

N89 - 10080

161042
R-17

AN EXPERT SYSTEM APPROACH TO ASTRONOMICAL DATA ANALYSIS

Mark D. Johnston

Space Telescope Science Institute¹
3700 San Martin Drive
Baltimore, MD 21218

SW 621003

ABSTRACT: *Expert System technology has much to offer to the problem of astronomical data analysis, where large data volumes and sophisticated analysis goals have caused a variety of interesting problems to arise. This paper reports the construction of a prototype expert system whose target domain is CCD image calibration. The prototype is designed to be extensible to different and more complex problems in a straightforward way, and to be largely independent of the details of the specific data analysis system which executes the plan it generates.*

I. INTRODUCTION

In recent years there has been an enormous increase in astronomical data volume and variety. A large fraction of astronomical data analysis is now automated, due to the increasing use of large format digital detectors on groundbased telescopes as well as data from satellite observatories. This trend will accelerate in the future with the launch of Hubble Space Telescope and the others in the "Great Observatories" series of satellites. At the same time, data analysis goals and methods have become increasingly sophisticated. New data reduction tools and techniques have been developed and are now in common use.

In response to the flood of data at all wavelengths, and to advances in display and computational hardware technology, astronomical data analysis systems have grown in number and functionality. The general philosophy of these systems is typically similar to the philosophy of the major computer operating systems: there is a command language (CL) which serves as the user interface in a "command/prompt" mode. The CL executes either single commands interactively, or scripts (procedures) of sets of commands (generally with a choice of interactive or batch/background execution). CL commands reduce to the execution of mod-

¹Operated by the Association of Universities for Research in Astronomy for the National Aeronautics and Space Administration

An Expert System Approach to Astronomical Data Analysis

ular operators which work on standardized types of data files. The advantages of this philosophy are clear:

- great flexibility for the user: individual commands can be combined to construct powerful tailored procedures
- ease of development: it is often relatively straightforward to add new modules, following a recipe that varies from system to system. Many programmers may thus independently contribute to the growth of a system.

There are, however, some serious drawbacks. Learning a system is not easy: commands are often complex with many parameters, and experts don't know all parts of even one system. To compound the problem, users often have to learn more than one system depending on where and how they obtain their data. It has also proven very difficult to capture and make available expert knowledge. Users can obtain assistance from manuals (often of enormous volume, hard to use and maintain), online help (often either irrelevant or a deluge of details), or by befriending the local expert on a particular topic. The same standardization that makes it easy for an expert to add new programs is often daunting to a casual user, who therefore writes "throwaway" code to accomplish some specific task. In some sense, the general philosophy breaks down when the size of a system becomes too large. In contrast to computer operating systems, where a novice need only learn a few commands to successfully navigate a system, there is no avoiding the complexity of a powerful analysis system when a difficult reduction is undertaken.

In response to these problems it is appropriate to consider alternative approaches to the general problem of astronomical data analysis. The most promising of these alternatives centers on the use of Expert System (ES) technology, which has matured dramatically in the past decade. Expert System languages and environments offer significant advantages over classical ones. They facilitate the manipulation of symbolic data, in contrast to languages that are primarily numerical in emphasis. They offer new data representation and programming paradigms that have proven successful on a wide variety of problems. Among these are:

- production (rule) systems: collections of IF...THEN... rules
- object-oriented programming: data representation in terms of modular "objects" (data structures) with control by "message passing" instead of procedure calls
- frames and inheritance: hierarchical data structures with properties inherited from parent classes
- new techniques for searching large problem spaces
- natural language capabilities: input/output in English-like syntax
- nonprocedural program specification: what to do rather than how to do it

These new programming methodologies allow the construction of programs that would not have been attempted before, and at the same time they increase programmer productivity. ES technology is maturing

An Expert System Approach to Astronomical Data Analysis

and becoming commercial: in recent years the field has developed from an area of computer science research into a variety of commercial products. The most important of these is the "Expert System Shell", which provides a general collection of inference mechanisms, data structures, and development tools that can be easily adapted for different applications. Examples now exist of applications of expert systems to a wide variety of problem types (see, e.g., [1,2]).

Other S/W engineering advances are integrated with current ES shells but not yet in most other computing environments. These advances include rapid prototyping as a system development methodology and facilities for easy user interaction with the system. In current expert system shells, advanced development tools and user interface features are integral. These include extensive graphics for display of system status and data, mouse/menu user interaction, multiple screen windows, and natural language command structures. Capabilities such as these make these systems ideal for interactive applications.

What can ES offer to development of astronomical data analysis?

- new programming methodologies to apply to current problems
- a way to capture and disseminate expert knowledge
- a powerful means to attack new problems

To date, application of ES technology to astronomical data have been limited to a few types of classification problems [3,4], but it is clear that the potential of ES methodologies will allow much more powerful and general applications.

The concept of an expert "Data Analysis Assistant" suggests vastly different things to different astronomers, ranging from sophisticated online help to what essentially amounts to an astronomical research assistant. For the present project, the following guiding principles were adopted:

- take advantage of the power of current data analysis systems wherever possible ("don't reinvent the wheel")
- take advantage of the development and user interface capabilities of ES shells running on the current generation of AI workstations. This leads to an architecture (discussed more fully below) where the ES resides on a workstation networked to a host; the latter can execute data analysis and display operations at the request of the workstation.
- deal first with the simpler problem of reasoning about data descriptions (rather than about data contents). This allows "loose coupling" of the workstation and host.
- extend the scope of the system to –
 - (1) reason about data contents,

An Expert System Approach to Astronomical Data Analysis

- (2) fully incorporate the user and host in the plan/execute/analyze cycle, and
- (3) reason about astronomical objects where this can productively help to guide the analysis process

This paper describes the construction and operation of a prototype system following the approach outlined above. The primary goals of the prototype were:

- develop a general means for representing knowledge about different kinds of data, instrument modes, and data analysis operations
- demonstrate the ability to construct a plan to achieve high-level analysis goals in terms of low-level operations, independent of any specific analysis system or language
- demonstrate the ability to efficiently recognize and eliminate redundant operations
- demonstrate the ability to automatically generate a command procedure to execute the plan in one of several specific data analysis languages
- include intrinsic extensibility to handle increasingly high-level goals and alternative plan generation strategies
- provide an easy-to-use mouse/graphics interface to the system user

Section II provides a description of the example problem chosen as a test domain, followed by a description of the design and operation of the prototype system. Plans for extension of the prototype are discussed in Section III, and some general conclusions are presented in Section IV.

II. THE PROTOTYPE

A. The Problem Domain

The problem chosen as the test domain was that of CCD calibration (sometimes called "pre-reduction") which consists of removing the gross instrumental signature from the astronomical CCD images. This is a tedious but relatively routine (at least in its basic form) preliminary step in the overall reduction of CCD images. This domain was chosen not because it represents an outstanding problem in astronomical data reduction but because it provides a simple test case for the methodology.

As implemented in the demonstration system there are four basic steps:

1. extraction of a subimage representing valid data
2. bias subtraction

An Expert System Approach to Astronomical Data Analysis

3. dark current correction

4. division by a flat field to correct for spatial nonuniformities in the CCD response

The first two steps depend only on characteristics of the instrument mode. The last two are more complicated, since dark and flat images are typically taken before and after science images and must therefore be identified and averaged to derive appropriate calibration images.

The operation of the prototype proceeds as follows:

1. The user provides a description of all relevant images (dark images, flat field images, and science images) to the system. In principle, most of this descriptive data could come from headers recorded with the image data, but there are some problems with this in practice.

2. The system analyzes the data descriptions to determine which dark and flat field images should be used to calibrate which science images. This step may identify science images for which no calibration data can be found: these are simply marked as problem images and ignored in subsequent processing.

3. The system then generates a plan network to calibrate each science image. This network consists of "tasks" representing image processing operations to be performed on the calibration and science images, along with the specification of any required ordering of the tasks. Any problems encountered in generating the plan are recorded and presented to the user.

4. Following plan generation, the user selects a specific command language for representing the plan. In the prototype two choices are offered: SDAS/IRAF (developed at Space Telescope Science Institute and National Optical Astronomy Observatories [5,6]) or MIDAS (developed at European Southern Observatory [7]). The system then turns the plan into a specific sequence of image processing commands (essentially a command procedure) in the chosen language. This procedure could simply be shipped via the network to the host for execution, although this was not implemented in the prototype.

B. Architecture and Implementation Hardware and Software

The prototype was constructed at the Space Telescope European Coordinating Facility at the European Southern Observatory in Munich. The system was implemented using the KEE Expert System shell from Intellicorp Inc., running on a Symbolics 3620 lisp machine; the latter will eventually be networked to a cluster of VAX 8600s which will serve as the host for image processing operations. A schematic view of the system architecture is shown in Figure 1. The user interacts only with the ES processor except for image display and manipulation which will be handled by the host processor. From the point of view of the

An Expert System Approach to Astronomical Data Analysis

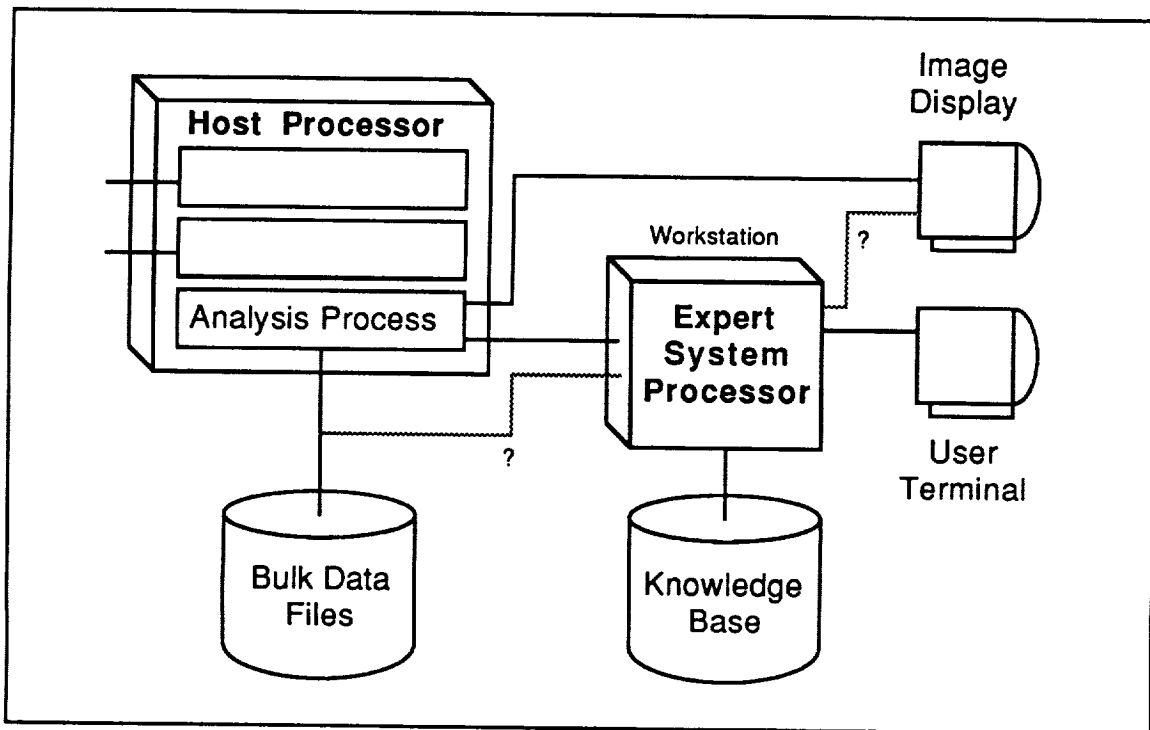


Figure 1—Architecture Overview. The host processor executes data analysis operations at the request of the ES processor. Bulk data files remain on the host, while the knowledge base is maintained on the ES workstation. Gray lines represent possible future connections whereby the ES could access bulk data files via a network file server, and directly control the image display.

host, the ES is simply another user accessing the system over the network. The gray lines represent possible future connections that would allow the ES processor to directly access bulk data files via a network file server, and to directly control graphics and image display devices. These are of particular interest if the ES can run efficiently on a general purpose workstation as well as on a lisp machine. This possibility will be evaluated on a Sun 3/160 workstation.

KEE provides a wide range of programming tools and techniques for implementing expert systems (indeed, one problem is to decide which of several possible representations or approaches are “best” for any particular problem.) Those of most relevance to the design of the prototype are the following:

production rules: rule classes may be defined and invoked in forward or backward chaining mode, or a combination of both

An Expert System Approach to Astronomical Data Analysis

frame system: this provides for the definition of classes of data objects which inherit properties from their parent classes. Frames have "slots" which can hold data items, references to other frames, procedures, etc. Of particular utility are "methods", which provide for object-oriented programming (procedures are invoked by sending messages from one object to another), and "active values", which are methods automatically invoked when a slot is set or referenced.

graphics interface: various kinds of graphical display or control images can be attached to frames and slots. These provide for method invocation when the user clicks with the mouse on an active region, graphical displays of system status, etc.

These features are well integrated, allowing for rapid construction and modification of a prototype.

C. System Design

The design of the prototype may be logically divided into the following areas:

1. knowledge representation (data, instrument modes, analysis tasks)
2. control structure
3. reasoning about the data
4. generation of the analysis plan
5. conversion of the plan into a language-specific procedure

Each of these areas is discussed in more detail in this section.

(1) Knowledge representation

A diagram representing the structure of the knowledge base is shown in Figure 2. Only those frames required for the operation of the prototype on the CCD calibration domain are shown, along with an indication of where extension would be possible to handle other types of problems. There are three major classes of objects:

Data: this class holds descriptions of various types of data. Only a few classes of CCD data are defined in the prototype: moving down the hierarchy, each class is a specialization of its parent class and contains slots that characterize that specific type of data. So, for example, the CCD-data class defines properties common to all CCD images (such as number of rows and columns), while the CCD-science class defines properties not shared by CCD-darks and CCD-flats (such as target name). Specific image files are made known to the system by defining a frame as a *member* of the appropri-

An Expert System Approach to Astronomical Data Analysis

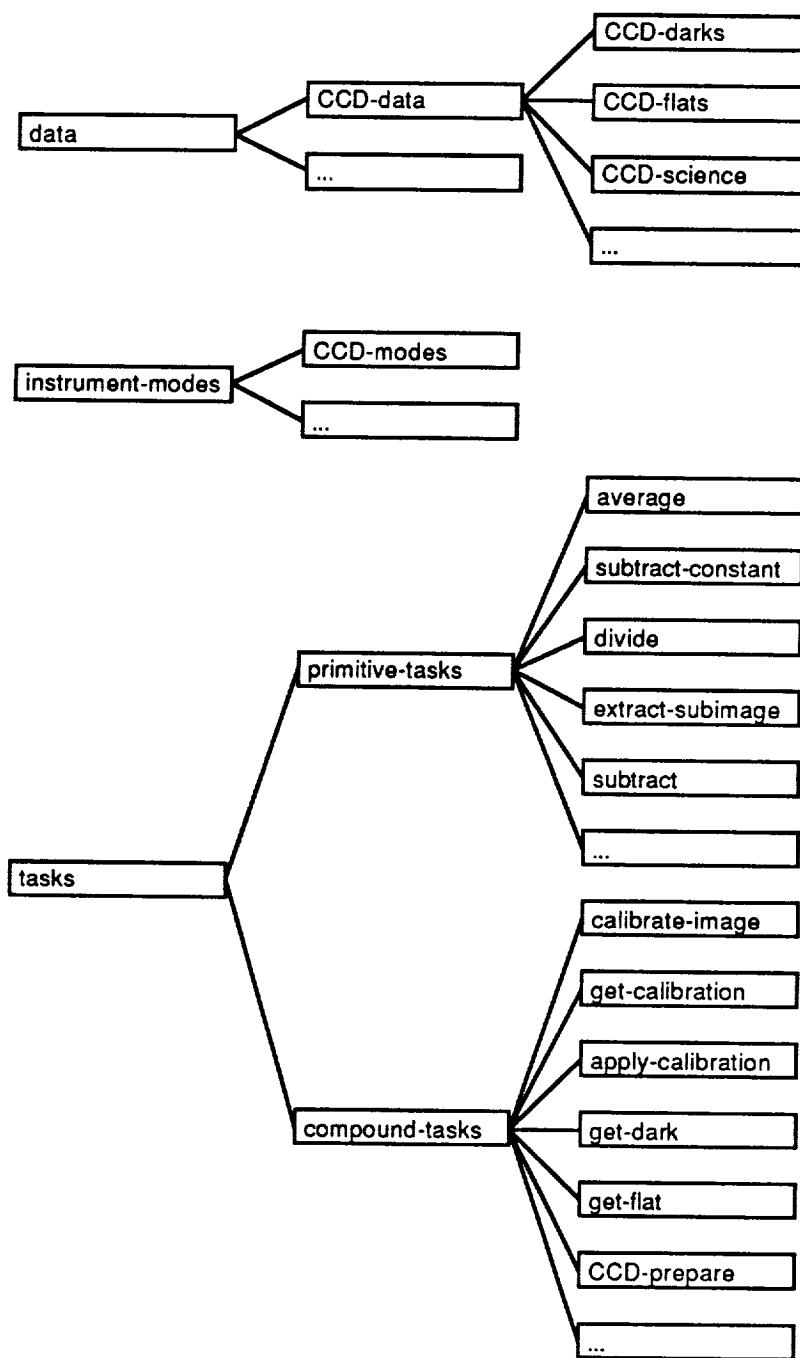


Figure 2.— Diagram representing the major classes defined in the prototype knowledge base (see §II C)

An Expert System Approach to Astronomical Data Analysis

ate class and by filling in its particular descriptive slots with values which characterize the image.

Instrument Modes: only a single subclass is defined in the prototype, representing a "pseudo-CCD" mode. Specific modes are defined as members of this class. Their slots hold such information as the number of valid rows and columns, the bias value to subtract, and any other properties which are common to all data taken in that mode. Members of the CCD-data class must reference a member of the CCD-modes class in order for the system to know mode-dependent data characteristics. This eliminates the need to repetitively specify mode characteristics in the frames which represent data.

Analysis Tasks: each task class represents a "generic" analysis operation (e.g. *calibrate-science-image*). A member of task classes represents a "specific" operation, i.e. with input and output files fully specified (e.g. *calibrate-frame-1*). The construction of the analysis plan involves the definition of these task members and linking them in the proper order, as described below.

Tasks are divided into primitive and compound tasks as follows:

primitive tasks are "atomic" in the sense that they can be implemented with a single command (or simple series of commands) in any specific data analysis system. Basic image arithmetic operations fall in this category, along with operations such as extracting a subimage of a larger image. Only primitive tasks need have any knowledge of how they are represented in a specific data analysis language. To accomplish this, each primitive task has defined a "format" method which can generate a command statement implementing the task. This method generates different command statements depending on the language currently selected by the user. It is important to note that this "format" method is the *only* place in the system where language-dependent information is required.

compound tasks represent "high-level" operations. Compound tasks cannot be directly converted into analysis commands, but instead must be "expanded" into a network of subtasks, each of which can be either compound or primitive. Ultimately, all compound tasks must expand into primitive tasks for the plan generation process to successfully complete. For example, the *calibrate-image* task expands into *get-calibration* and *apply-calibration*; the former expands into *get-dark* and *get-flat*, etc. An important design feature is that compound tasks need only specify tasks into which they immediately expand. This makes it easy to add new high-level tasks by using those already defined as building blocks.

An Expert System Approach to Astronomical Data Analysis

(2) Control strategy

The operation of the prototype is controlled by the user by clicking in active regions of a graphical "control panel" (see Figure 3 for a copy of the screen immediately after a "reset" operation has completed). In the prototype the user can only take four actions (once the data descriptions have been entered):

1. reset: initialize the system
2. run: invoke the plan generation step
3. choose language: one of SDAS/IRAF or MIDAS must be selected
4. format: convert the plan generated by step (3) into a command procedure in the selected language

The plan generation step is controlled by a set of forward-chaining production rules. Conflict resolution is controlled by grouping rules by "salience" or weight; within a group, recency of rule instantiation determines which rule to fire next. The major salience groups are (in priority order):

1. reason about data properties
2. check for problems with data
3. establish top-level goals
4. merge redundant tasks
5. expand compound tasks
6. check for problems with tasks

Each of these groups is discussed below with the exception of (3): this group contains only a single rule to generate a *calibrate-science-image* task for each uncalibrated science image. In a more general system, the goal (or goals) would be specified by the user rather than by a specific rule of this type.

The plan generation process constructs a directed acyclic graph of tasks representing the analysis plan. Each task in this network contains references to any tasks that must immediately precede or follow it. These task-to-task "links" are used to ensure that any required task orderings are preserved as the plan evolves, without committing early to a linear sequence of tasks that would be hard to reorder later. While it is possible to manage these links with a rather complex set of rules, it is easiest to manage them procedurally via methods and message-passing. This approach greatly simplifies the rulebase and speeds up system operation, and also illustrates nicely how the integration of object-oriented and rule approaches can simplify system design and implementation.

(3) Reasoning about the data

The purpose of this rule class is:

- to infer any unspecified data properties that might be required when plan generation starts, and

An Expert System Approach to Astronomical Data Analysis

KEE Typescript Window

Resetting all tasks/images...

FLAT1 FLAT2 DARK1 DARK2 DARK3 SCIENCE1 SCIENCE2 SCIENCE3

Done.

ORIGINAL PAGE IS
OF POOR QUALITY

DATA STATUS DISPLAY PANEL

SCIENCE-1 CALIBRATION STATUS				Comments
NO	IN-PROGRESS	PROBLEM	COMPLETE	
Dark Frames		Flat Frames		DISPLAY
Unknown		Unknown		

SCIENCE-2 CALIBRATION STATUS				Comments
NO	IN-PROGRESS	PROBLEM	COMPLETE	
Dark Frames		Flat Frames		DISPLAY
Unknown		Unknown		

SCIENCE-3 CALIBRATION STATUS				Comments
NO	IN-PROGRESS	PROBLEM	COMPLETE	
Dark Frames		Flat Frames		DISPLAY
Unknown		Unknown		

MAIN CONTROL PANEL

RESET	RUN	FORMAT	# Rules Fired	# Unexpanded Comp.	# Expanded Compounds
0	0	0	0 25 50 75 100	0 10 20 30 40	0 10 20 30 40
COMMAND LANGUAGE			# Task Merges	# In-progress Comp.	# Primitive Tasks
SDAS MIDAS			0 5 10 15 20	0 10 20 30 40	0 10 20 30 40

Figure 3—What the screen looks like immediately after execution of a "reset" command. The lower panel is used for overall system control and status monitoring. The upper right panel displays the calibration status of three science images as the system runs. Informative messages appear in the upper left text window. The user clicks with the mouse on an active region (e.g. RESET) to control operation.

An Expert System Approach to Astronomical Data Analysis

- to associate calibration images (darks and flats) with science images.

In the present prototype this association is very simple (e.g., flat fields need only be taken on the same night as the science image with the same instrument mode and filters). There are also rules to detect and annotate problems, such as an inability to identify any calibration images for a particular science image. Extension of this rule class to handle more complex cases is straightforward.

(4) Generation of the analysis plan

Plan generation consists of two competing processes: **expansion** of compound tasks and **merging** of redundant tasks.

Expansion of compound tasks requires at least one rule per task; alternative expansion strategies would be implemented by including multiple rules which look for appropriate preconditions. An expanded compound task is marked as such, and all predecessor/successor links in which it took part are removed and attached to its newly created subtasks (see Figure 4(a) for an illustration). Link manipulation is handled procedurally, by sending a message to a task that it should propagate all of its current links to its subtasks. Intermediate data files are automatically given unique names. Expanded tasks are not deleted: they are used for identifying mergeable tasks as described below.

Some tasks expand into a fixed number of subtasks, while others can generate an arbitrary number depending on the circumstances. For example, the *get-dark* task expands into an unordered series of *CCD-prepare* tasks (one for each input image) followed by a single *average* task. A minimum of three rules were found necessary for this situation: one to initiate expansion and create any fixed tasks (*average* in the *get-dark* case); one to create and link each of the variable number of subtasks (*CCD-prepare* in the current example), and a final rule which notes that expansion is complete.

Merging of redundant tasks is accomplished by one generic rule, which essentially says that if two tasks of the same type are found with identical input, then mark one as redundant and change the predecessor/successor links of all affected tasks to reflect the required ordering (see Figure 4(b,c) for an illustration of this process). A note is made in a table that the output files of the redundant task and its replacement are to be identified. As with task expansion, the maintenance of task link information is performed procedurally. The task merging rule has higher priority than any task expansion rule, in order to catch redundancies at as high a level as possible. In the prototype, two of the example science images use exactly the same set of dark and flat field images for calibration: as a result of the merging rule, the tasks which compute the average dark and flat field images will only be planned once.

An Expert System Approach to Astronomical Data Analysis

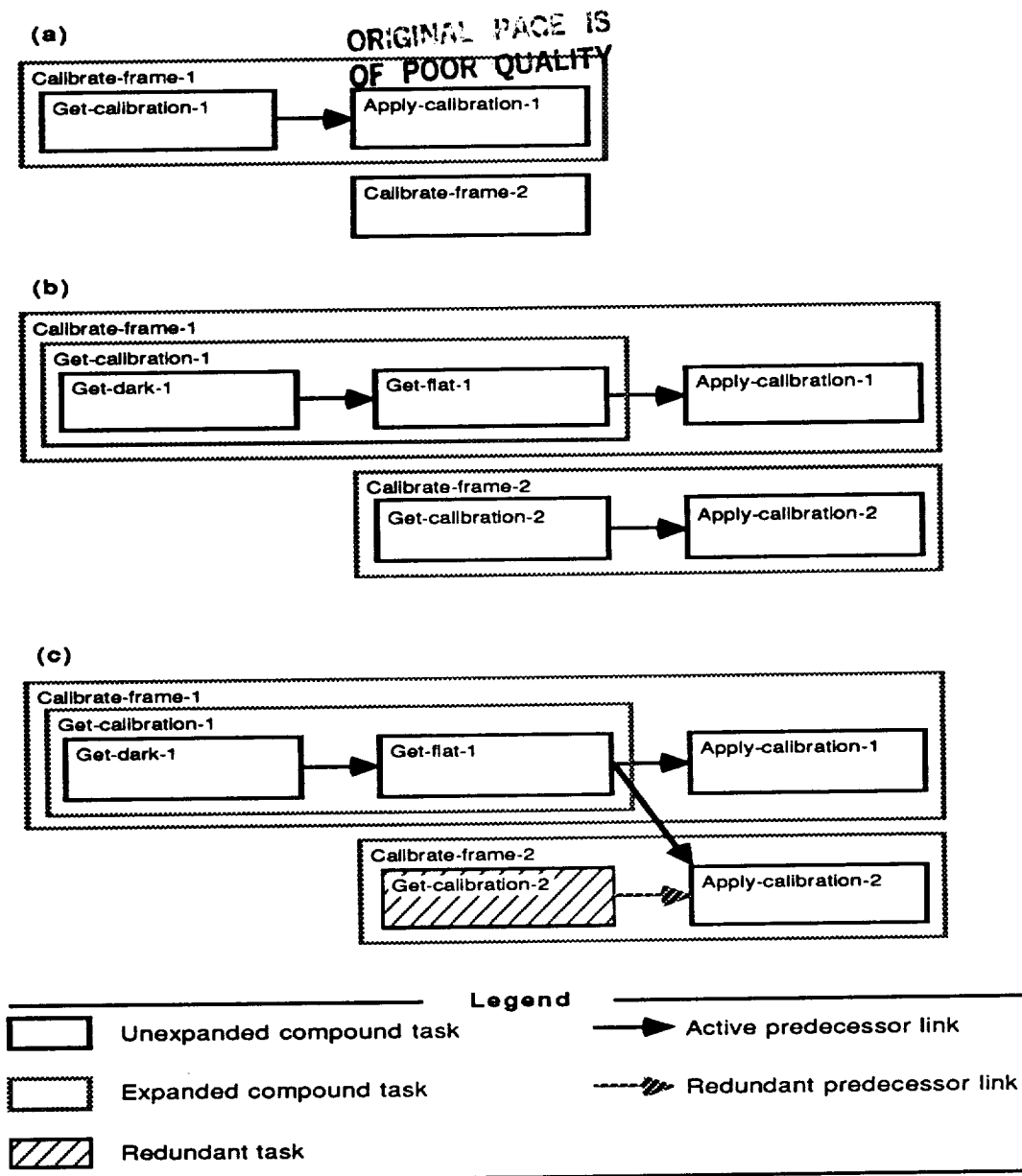


Figure 4— Illustration of the expansion and merging of tasks as the task network is built. (a) calibrate-frame-1 is expanded (b) calibrate-frame-2 is expanded, and the subtask get-calibration-1 of calibrate-frame-1 is expanded (c) the system recognizes get-calibration-2 as identical to get-calibration-1: it is therefore marked as redundant and the predecessor links are changed to ensure that the tasks remain properly ordered. See §II C(4).

An Expert System Approach to Astronomical Data Analysis

At the conclusion of expand/merge rule processing, any unexpanded compound tasks are marked as errors and displayed to the user. Their presence indicates that the rulebase is not complete and that no knowledge about how to accomplish these tasks resides in the system.

(5) Conversion of the plan into a language-specific procedure

Once created, the task network representing the analysis plan resides in the knowledge base until the user clicks on the "reset" active region. This network may be converted into a command procedure in one of the two currently supported languages, SDAS/IRAF or MIDAS. The conversion process is procedural and is initiated on user request: the procedure simply traverses the network and sends each primitive task a "format" message after ensuring that all of its predecessors have already been processed. No optimizing of the traversal is performed, but this would certainly be possible (for example, to minimize the disk space required at any one time for storage of intermediate files).

III. FUTURE DIRECTIONS

Although the prototype system accomplishes the goals listed in Section I, there are a number of areas which require extension before a useful system could be considered operational even in the limited domain of CCD calibration. This section first addresses the shortcomings of the current prototype, then discusses directions in which the prototype should be extended.

- The current system does not represent the effects of analysis operations on the data. Intermediate files are represented simply as symbolic names. This means that it is not possible to write rules that reference the *properties* of partially processed data in their preconditions. Adding this capability will require the creation of frames with slots that describe how a particular analysis task modifies the data in an image file. Each intermediate data file would be represented by such a frame.
- The current rules for associating calibration images with science images are trivial. However, it would be straightforward to incorporate more realistic criteria for this process.
- There are no tasks defined to fix cold columns and perform other data repair operations. These can, however, be straightforwardly added to the current structure.
- Graphics displays or system status are currently limited to the specific science frames that are part of the test dataset. It would be useful and not difficult to allow the creation of monitoring graphics "on the fly" at user request.

An Expert System Approach to Astronomical Data Analysis

The current prototype will be extended by incorporating rules representing more realistic calibration criteria, and by exercising the system on a real collection of science and calibration images.

There are several topics that must be addressed for the prototype to be useful beyond its current limited domain:

- **reasoning about data contents:** at present the system is based entirely on reasoning based on a *description* of data files, but this is clearly inadequate for more involved problems (and even to handle certain subtleties associated with the CCD calibration domain). It is essential to include a capability to obtain information about the *contents* of data files to proceed further in this direction. This will require one of two approaches: most desirable is the ability of the ES to initiate realtime tasks on the host processor to compute and return information about the data to the ES, such as statistics on pixel values, the results of fitting trial point-spread functions, etc. Alternatively, it may be possible to iterate with the host by preparing a batch procedure to generate this information; the ES would then have to preserve its current state until the results return.
- **user interaction:** the experienced eye of an astronomer is undoubtedly essential in many aspects of astronomical data reduction. While this is related to the problem of reasoning about data contents, it adds the further complication of ensuring that the user may, upon demand, display raw and intermediate files and provide input to be used by the ES in subsequent processing.
- **goal specification:** planning an analysis process is strongly driven by the ultimate goals of the analysis, in contrast to general CCD calibration domain where the same steps are followed for virtually all planned uses of the data. A facility must be provided to allow the astronomer to specify high level goals which will then be used by the ES to guide further planning.
- **reasoning about astronomical objects:** the prototype system contains no knowledge of the properties of astronomical objects, but this is clearly an area that would be extremely fruitful to pursue.
- **practical issues:** certain practical problems must be solved before productive use of ES can be expected to become routine. For example, the speed of rule processing tends to be slower than procedural code and may become a serious limitation for a large operational system. Another problem is that certain data descriptors that are required to automate processing are not recorded in a standardized way at all telescope sites (this problem is not peculiar to ES processing!).

IV. CONCLUSIONS

The prototype described in this paper demonstrates the feasibility of applying expert system technology to the problem of astronomical data analysis. The major goals of the prototype were accomplished in a rela-

An Expert System Approach to Astronomical Data Analysis

tively short time by taking advantage of the unique capabilities of current commercially available hardware and software systems. It is clear that the approach holds great promise for the future. Major problems to be addressed by the continuation of this project include:

- processing of large volumes of data via intelligent automated analysis; of particular interest is the integration of ES with modern pattern recognition and object classification techniques
 - exploiting the expected properties of data to guide the development and execution of reduction procedures
 - automation of tedious but necessary steps in data reduction
 - minimizing the problem of proliferation of analysis systems and languages by providing a language-independent high-level user interface
 - capturing techniques used by experts and making them available to less experienced users
-

The author is grateful for the hospitality and support of the Space Telescope European Coordinating Facility (ST-ECF) and the European Southern Observatory (ESO) where this work was conducted. Particular mention should be made of Rudi Albrecht, Hans-Martin Adorf, Roberto Rampazzo, and Dietrich Baade (ST-ECF), Glenn Miller and Mike Shara (Space Telescope Science Institute), and Don Rosenthal (STScI and NASA Ames Research Center, Intelligent Systems Division) who have provided many useful comments and suggestions on this work. Intellicorp GmbH (Munich) provided prompt technical support for the KEE system, and Symbolics GmbH (Eschborn/Ts) provided access to the lisp machine.

An Expert System Approach to Astronomical Data Analysis

REFERENCES

1. *Building Expert Systems*, ed. Hayes-Roth, F., Waterman, D., and Lenat, D., (Addison-Wesley: 1983)
2. *Rule-Based Expert Systems*, ed. Buchanan, B., and Shortliffe, E., (Addison-Wesley: 1985)
3. Thonnat, M., Granger, C., and Berthod, M., IEEE Transactions on Computer Vision and Pattern Recognition, 206-208 (1985)
4. Bernat, A., and McGraw, J., Steward Observatory Preprint No. 694 (1986)
5. *Science Data Analysis Software User's Manual*, Space Telescope Science Institute 86053A (1986)
6. Shames, P., and Tody, D., "A User's Guide to the IRAF Command Language", Space Telescope Science Institute SDAS/IRAF Workshop (December 1986)
7. Crane, P., Banse, K., Grosbøl, P., Ounnas, C., and Ponz, D., "MIDAS", in *Data Analysis in Astronomy*, ed. di Gesù et al. (Plenum Press: 1985)

Section D

Expert System Tools/Techniques

